

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application. Please amend the claims, as follows.

1. (Currently Amended) A method for replicating one or more data objects from a source system to a target system, the method comprising:

~~creating~~ providing an electronic data element comprising a first data field and a second data field, wherein the first data field contains data representing an identifier assignable to the one or more data objects and the second data field contains data representing a state of the identifier, the second data field configured to store one of:

a) a first state, in which said electronic data element may be accessed by one or more data object processing operations and whereby said identifier is assignable to one or more data objects,

b) a second state, in which said electronic data element may not be accessed by one or more data object processing operations and whereby said identifier is assignable to one or more data objects by one or more data object processing operations having already accessed said identifier at a time when said identifier was in the first state, or

c) a third state, in which said electronic data element may not be accessed by one or more data object processing operations and whereby said identifier is not assignable to one or more data objects;

assigning the identifier to the one or more data objects;

processing the one or more data objects in accordance with a software application;

storing the one or more processed data objects on the source system;

changing the state of the identifier in the electronic data element to indicate that the one or more processed data objects are ready to be replicated from the source system to the target system; and

replicating, in response to changing the state of the identifier, the one or more processed data objects from the source system to the target system.

2. (Canceled)

3. (Original) The method of claim 1, wherein the first data field and the second data field are in a table.

4. (Original) The method of claim 1, wherein the first data field is in a first table and the second data field is in a second table.

5. (Original) The method of claim 1, wherein the electronic data element is implemented in object orientated programming as an instance of a class.

6. (Previously Presented) The method of claim 1, wherein the electronic data element further comprises a third data field configured to store data representing whether the identifier stored in the first data field is a default identifier.

7. (Original) The method of claim 2, further comprising changing the first data field from the first state to the second state.

8. (Canceled)

9. (Previously Presented) The method of claim 7, further comprising changing the second data field to the third state after the one or more processed data objects have been stored on the source system.

10. (Previously Presented) The method of claim 9, further comprising:

creating a new electronic data element comprising a first data field and a second data field, wherein the first data field contains data representing an identifier assignable to one or more data objects and the second data field contains data representing a state of the identifier; and

setting the second field of the new electronic data element to the first state.

11. (Previously Presented) The method of claim 10, wherein the new electronic data element further comprises a third data field configured to store data representing whether the identifier stored in the first data field of the new electronic data element is a default identifier, and the method further comprising storing data in the third data field of the new electronic data element to indicate that the first data field of the new electronic data element contains the default identifier.

12. (Previously Presented) The method of claim 10, further comprising setting the second field of the electronic data element to the second state.
13. (Previously Presented) The method of claim 6, further comprising changing the third data field of the electronic data element from a value corresponding to the default identifier to a value corresponding to an identifier other than the default identifier.
14. (Original) The method of claim 2, further comprising:
 - setting a block on the electronic data element;
 - examining the second field of the electronic data element; and
 - if the state of the second field of the electronic data element is the first state or the second state, preventing a change in the state of the second field to the third state.
15. (Previously Presented) The method of claim 14, removing the block if storing of the one or more data objects is committed.
16. (Original) The method of claim 14, further comprising irreversibly setting the block if the electronic data element is in the third state.
17. (Original) The method claim 1, further comprising share locking the electronic data element.

18. (Previously Presented) The method of claim 17, further comprising share locking the electronic data element prior to assigning the identifier of the electronic data element to the one or more data objects.

19. (Previously Presented) The method of claim 17, further comprising unlocking the share locking of the electronic data element after storing of the one or more data objects is committed.

20. (Previously Presented) The method of claim 17, further comprising examining the state of the share lock of the data element prior to assigning the identifier of the electronic data element to the one or more data objects.

21. (Canceled)

22. (Original) The method of claim 1, wherein the identifier of the first data field comprises a globally unique identifier.

23. (Original) The method of claim 1, wherein the identifier of the first data field comprises a time stamp.

24. (Currently Amended) A system for replicating one or more data objects from a source system to a target system , the system comprising:

a memory; and

a microprocessor coupled to the memory and programmed to:

~~create~~provide an electronic data element comprising a first data field and a second data field, wherein the first data field contains data representing an identifier assignable to the one or more data objects and the second data field contains data representing a state of the identifier, the second data field configured to store one of:

a) a first state, in which said electronic data element may be accessed by one or more data object processing operations and whereby said identifier is assignable to one or more data objects,

b) a second state, in which said electronic data element may not be accessed by one or more data object processing operations and whereby said identifier is assignable to one or more data objects by one or more data object processing operations having already accessed said identifier at a time when said identifier was in the first state, or

c) a third state, in which said electronic data element may not be accessed by one or more data object processing operations and whereby said identifier is not assignable to one or more data objects;

assign the identifier to the one or more data objects;

process the one or more data objects in accordance with a software application;

store the one or more processed data objects on the source system;

change the state of the identifier in the electronic data element to indicate that the one or more processed data objects are ready to be replicated from the source system to the target system; and replicate, in response to changing the state of the identifier, the one or more processed data objects from the source system to the target system.

25. (Canceled)

26. (Original) The system of claim 24, wherein the first data field and the second data field are in a table.

27. (Original) The system of claim 24, wherein the first data field is in a first table and the second data field is in a second table.

28. (Original) The system of claim 24, wherein the electronic data element is implemented in object orientated programming as an instance of a class.

29. (Previously Presented) The system of claim 24, wherein the electronic data element further comprises a third data field configured to store data representing whether the identifier stored in the first data field is a default identifier.

30. (Original) The system of claim 25, wherein the microprocessor is further programmed to change the first data field from the first state to the second state.

31. (Canceled)

32. (Previously Presented) The system of claim 30, wherein the microprocessor is further programmed to change the second data field to the third state after the one or more processed data objects have been stored on the source system.

33. (Previously Presented) The system of claim 32, wherein the microprocessor is further programmed to:

create a new electronic data element comprising a first data field and a second data field, wherein the first data field contains data representing an identifier assignable to one or more data objects and the second data field contains data representing a state of the identifier; and

set the second field of the new electronic data element to the first state.

34. (Previously Presented) The system of claim 33, wherein the new electronic data element further comprises a third data field configured to store data representing whether the identifier stored in the first data field of the new electronic data element is a default identifier, and wherein the microprocessor is further programmed to store data in the third data field of the new electronic data element to indicate that the first data field of the new electronic data element contains the default identifier.

35. (Previously Presented) The system of claim 33, wherein the microprocessor is further programmed to set the second field of the electronic data element to the second state.

36. (Previously Presented) The system of claim 29, wherein the microprocessor is further programmed to change the third data field of the electronic data element from a value corresponding to the default identifier to a value corresponding to an identifier other than the default identifier.

37. (Original) The system of claim 25, wherein the microprocessor is further programmed to:

- set a block on the electronic data element;
- examine the second field of the electronic data element; and
- if the state of the second field of the electronic data element is the first state or the second state, prevent a change in the state of the second field to the third state.

38. (Previously Presented) The system of claim 37, wherein the microprocessor is further programmed to remove the block if storing of the one or more data objects is committed.

39. (Original) The system of claim 37, wherein the microprocessor is further programmed to irreversibly set the block if the electronic data element is in the third state.

40. (Original) The system claim 24, wherein the microprocessor is further programmed to share lock the electronic data element.

41. (Previously Presented) The system of claim 40, wherein the microprocessor is further programmed to share lock the electronic data element prior to assigning the identifier of the electronic data element to the one or more data objects.

42. (Previously Presented) The system of claim 40, wherein the microprocessor is further programmed to unlock the share locking of the electronic data element after storing of the one or more data objects is committed.

43. (Previously Presented) The system of claim 40, wherein the microprocessor is further programmed to examine the state of the share lock of the data element prior to assigning the identifier of the electronic data element to the one or more data objects.

44. (Canceled)

45. (Original) The system of claim 24, wherein the identifier of the first data field comprises a globally unique identifier.

46. (Original) The system of claim 24, wherein the identifier of the first data field comprises a time stamp.

47. (Previously Presented) The method of claim 1, wherein the software application is a business application.

48. (Previously Presented) The method of claim 47, wherein the business application is an enterprise resource planning software application.

49. (Previously Presented) The method of claim 47, wherein the step of replicating the one or more data objects from the source system to the target system is performed by a software application other than the business application.

50. (Previously Presented) The method of claim 1, wherein the one or more data objects includes at least one booking, accounting, invoicing, receipt, or voucher data object.

51. (Previously Presented) The system of claim 24, wherein the software application is a business application.

52. (Previously Presented) The system of claim 51, wherein the business application is an enterprise resource planning software application.

53. (Previously Presented) The system of claim 51, wherein the microprocessor executes a software process other than the business application to replicate the one or more data objects from the source system to the target system.

54. (Previously Presented) The system of claim 24, wherein the one or more data objects includes at least one booking, accounting, invoicing, receipt, or voucher data object.

55. (Previously Presented) The system of claim 24, wherein the microprocessor comprises one or more processors.

56. (Currently Amended) A system for replicating one or more data objects from a source system to a target system, the system comprising:
means for ~~creating~~ providing an electronic data element comprising a first data field and a second data field, wherein the first data field contains data representing an identifier assignable to the one or more data objects and the second data field contains data representing a state of the identifier, the second data field configured to store one of:

a) a first state, in which said electronic data element may be accessed by one or more data object processing operations and whereby said identifier is assignable to one or more data objects,

b) a second state, in which said electronic data element may not be accessed by one or more data object processing operations and whereby

said identifier is assignable to one or more data objects by one or more data object processing operations having already accessed said identifier at a time when said identifier was in the first state, or

c) a third state, in which said electronic data element may not be accessed by one or more data object processing operations and whereby said identifier is not assignable to one or more data objects;

means for assigning the identifier to the one or more data objects;

means for processing the one or more data objects in accordance with a software application;

means for storing the one or more processed data objects on the source system;

means for changing the state of the identifier in the electronic data element to indicate that the one or more processed data objects are ready to be replicated from the source system to the target system; and

means for replicating, in response to changing the state of the identifier, the one or more processed data objects from the source system to the target system.

57. (Previously Presented) The system of claim 56, wherein the software application is a business application.

58. (Previously Presented) The system of claim 57, wherein the business application is an enterprise resource planning software application.